PFM-CIS-VI-001



UNIVERSIDAD
NACIONAL
DE LOJA



Área de la Energía, las Industrias y los Recursos Naturales No Renovables

---

Carrera de Ingeniería en Sistemas

Lenguaje Ensamblador

Sexto A

Autor:

- Bravo-Linder

Docente:

- Ing. Guamán-René

FECHA:

- 19/05/2014

Loja - Ecuador
2014

Índice

1. UNIDAD DE EJECUCIÓN Y UNIDAD DE INTERFAZ DEL BUS

EL procesador se divide en dos unidades lógicas: una unidad de ejecución(EU) y una unidad de interfaz del bus(BIUS) como se ilustra en la siguiente figura. EL papel de la EU es ejecutar instrucciones, mientras que la BIU envía instrucciones y datos a la EU. La EU contiene una unidad aritmética-lógica(ALU),una unidad de control(CU) y varios registros. Estos elementos ejecutan instrucciones y operaciones aritméticas y lógicas.
La función mas importante de la BIU es manejar la unidad de control del bus, los registros de segmentos y la cola de instrucciones. La BIU controla los buses que transfieren los datos a la EU, a la memoria y a los dispositivos de entrada/salida externos, mientras que los registros de segmentos controlan el direccionamiento de memoria.[1]

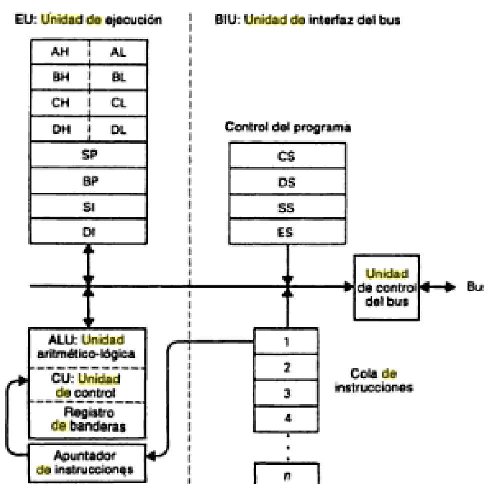


Figura 1. Unidad de ejecución y unidad de interfaz del bus

Otra función de la BIU es permitir el acceso a instrucciones. Ya que las instrucciones de un programa en ejecución se encuentran en la memoria, la BIU debe accesar instrucciones desde la memoria y colocarlas en la cola de instrucciones. Puesto que el tamaño de esta cola es de 4 a 32 bytes, dependiendo del procesador, la BIU es capaz de adelantarse y buscar con anticipación instrucciones de manera que siempre haya una cola de instrucciones listas para ser ejecutadas.
La EU y la BIU trabajan en paralelo, si bien la BIU se mantiene un paso adelante. La EU notifica a la BIU cuando necesita acceso a los datos en memoria o a un dispositivo de E/S.
También, la EU solicita instrucciones de máquina de la cola de instrucciones de la BIU. La instrucción que se encuentra adelante de la cola es la actualmente ejecutable, y mientras la EU está ocupada ejecutando una instrucción, la BIU busca otra en la memoria. Esta búsqueda se traslapa con la ejecución y aumenta la velocidad de procesamiento.
Los procesadores hasta el 80486 tienen lo que se conoce como tubería sencilla, la cual los restringe a completar una instrucción antes de iniciar la siguiente. El pentium y procesadores posteriores tienen un tubería doble (o dual) que les permite correr varias operaciones

en paralelo.[1]

2. MEMORIA INTERNA (ROM Y RAM)

La microcomputadora posee dos tipos de memoria interna: memoria de acceso aleatorio(RAM) y memoria de solo lectura(ROM). Los bytes en memoria se numeran en forma consecutiva, iniciando con 00, de modo que cada localidad tiene un número de dirección único.[1]

2.1. ROM

La ROM es un chip especial de memoria que(como su nombre lo indica) sólo puede ser leída. Ya que las instrucciojnes y los datos están grabados permanentemente en un chip de ROM, no pueden ser alterados. El sistema básico de Entrada/Salida(BIOS) de ROM inicia en la dirección 768k y maneja los dispositivos de entrada/salida, como un controlador de disco duro. La ROM que inicia en 960k controla las funciones básicas de la computadora, como la autoprueba al encender, patrones de puntos para los gráficos y el autocargador de disco.Cuando se enciende la computadora, la ROM realiza ciertas verificaciones y carga, desde el disco, los datos especiales del sistema que envía a la RAM.[1]

Inicio Dirección Dec	Hex	Uso	
960K	F0000	64K sistema base de ROM	memoria superior
768K	C0000	192K área de expansión de memoria (ROM)	
640K	A0000	128 K área de despliegue de video (RAM)	
cero	00000	640 K memoria (RAM)	memoria convencional

Figura 1-3 Mapa de memoria física

Figura 2. Mapa de memoria física

2.2. RAM

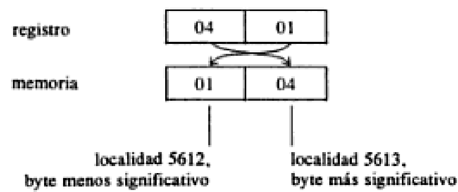
Un programador está preocupado principalmente con la RAM, que sería mejor llamada memoria de lectura-escritura. La RAM se dispone como una hoja de trabajo para almacenamiento temporal y ejecución de programas.

Ya que el contenido de la RAM se pierde cuando se apaga la computadora, debe reservar almacenamiento externo para guardar programas y datos. SI cuando enciende la computadora tiene insertado un disco flecible con DOS o un disco duro instalado, el procedimiento de arranque en ROM carga el programa COMMAND.COM en RAM. Después se le pide a COMMAND.COM realizar acciones, como cargar un programa de un disco a la RAM. Puesto que el COMMAND.COM ocupa una pequeña parte de RAM, también existe espacio para otros programas. Su programa se ejecuta en RAM y por lo común

produce salida a la pantalla, a la impresora o a un disco. Cuando termina, ested puede pedir al COMMAND.COM cargar otro programa en RAM, una acción que se escribe sobre el programa anterior.[1]

3. DIRECCIONAMIENTO DE MEMORIA

Dependiendo del modelo, el procesador puede accesar uno o mas bytes de memoria a la vez. Considere el número decimal 1,025. La representación hexadecimal de esta cifra, 0401H, requierede dos bytes(o una palabra) de memoria. Consta de un byte de orden alto(más significativo), 04, y un byte de orden bajo (menos significativo), 01.El sistema almacena en memoria estos bytes en secuencia inversa de bytes: el byte de orden bajo en la dirección baja de memoria y el byte orden alto en la dirección alta de memoria. Por ejemplo, el procesador transferiría 0401H de un registro a las localidades de memoria 5612 y 5613 como:



El procesador espera que los datos numéricos en la memoria estén en secuencia inversa de bytes y los procesa de acuerdo con esto. Cuando el procesador recupera la palabra de la memoria, otra vez invierte los bytes, restableciendo de manera correcta en el registro como 04 01 hex.
Aunque esta característica es enteramente automática usted tiene que estar alerta cuando programe y depure programas en lenguaje ensamblador.
Un programador de lenguaje ensamblador tiene que distinguir claramente entre la dirección y los contenidos de una localidad de memoria. En el ejemplo anterior, el contenido de la localidad 5612 es 01 y el contenido de la localidad 5613 es 04.[1]

4. Bibliografía

Referencias

[1] P.Abel, 1996, “ Lenguaje Ensamblador y Programación Para PC IBM y Compatibles”, Tercera edición, México, Editorial PRENTICE-HALL. Disponible en web: http://books.google.com.ec/books?id=syAOv7jF3W4C&hl=es&source=gbs_navlinks_s